Amendments to the Claims:

1. (currently amended) A computer system comprising:

a memory comprising:

a message center;

an interpreter for interpreting data in a multiple task system, the interpreter comprising:

a scanner for reading at least one command from an input port and providing a token according to the category of the command;

a first parser for interpreting the command when the token indicates the first parser; and

a second parser for interpreting the command when the token indicates the second parser, for temporarily storing data generated after interpreting the command into the message center in the memory, and for executing the data stored in the message center after interpreting all other corresponding commands; and

a processor for processing programs and data stored in the memory;

wherein the scanner is for providing the token indicating the first parser for application-level commands, and is for providing the token indicating the second parser for driver-level commands.

2. (original) The computer system of claim 1 wherein the interpreter further comprises a symbol table for providing a mapping of commands and symbols, in order to convert symbols input at the input port into commands.

3. (cancelled)

4. (original) The computer system of claim 1 wherein the input port is a command line

or a script file.

5. (original) The computer system of claim 1 wherein the first parser and the second parser can output data generated after interpreting the command by the first parser and the second parser to an output port.

6. (original) The computer system of claim 5 wherein the output port is a file stored in the memory.

7. (currently amended) A method for executing an interpreter code in a multiple task system in a computer system, wherein the computer system comprising a memory and a processor, the memory comprising a scanner, a first parser, and a message center, the method comprising:

reading at least one command from an input port by the scanner and providing a token according to the category of the command;

interpreting the command by the first parser when the token indicates the first parser;

storing a second parser into the memory; ~~and~~

interpreting the command by the second parser when the token indicates the second parser, temporarily storing data generated after interpreting the command into the message center in the memory, and executing the data stored in the message center after interpreting all other corresponding commands; and

providing the token indicating the first parser for application-level commands, and providing the token indicating the second parser for driver-level commands.

8. (original) The method of claim 7 wherein the scanner converts symbols inputted at the input port into commands according to a symbol table.

9. (cancelled)

10. (original) The method of claim 7 wherein the input port is a command line or a script file.

11. (currently amended) The method of claim 7 wherein the first parser and the second parser can output data generated after interpreting the command by the first parser and the second ~~parserto~~ parser to an output port.

12. (original) The method of claim 11 wherein the output port is a file stored in the memory.

13. (new) The computer system of claim 1 wherein the command(s) read by the scanner belong to an F-language.

14. (new) The computer system of claim 1 wherein the application-level commands include mathematical operations, stack operations, logical operations, and printer commands.

15. (new) The computer system of claim 1 wherein the driver-level commands include input output I/O actions, memory access, and call interruptions.

16. (new) The computer system of claim 1 wherein the driver-level commands are commands requiring processing from drivers provided by an operating system of the multiple task system.

17. (new) The method of claim 7 for executing the interpreter code being of an F-language.

18. (new) The method of claim 7 wherein the application-level commands include mathematical operations, stack operations, logical operations, and printer commands.

19. (new) The method of claim 7 wherein the driver-level commands include input output I/O actions, memory access, and call interruptions.

20. (new) The method of claim 7 wherein the driver-level commands are commands requiring processing from drivers provided by an operating system of the multiple task system.